

**Gillis, Brian J (AU2141)**

---

**From:** Bernard Tomsa [btomsa@brookskushman.com]  
**Sent:** Thursday, October 29, 2009 2:56 PM  
**To:** Gillis, Brian J (AU2141)  
**Subject:** 10/725,738

**Attachments:** 10 725 738 Claims.doc



10 725 738  
Claims.doc (54 KB)

Brian,

I apologize for the delay in getting this to you. Attached is an amendment that should be suitable for use as an examiner's amendment, and change all claims in the manner suggested by you in our conversation last Thursday, October 22, 2009. Please let me know if additional changes are needed.

Bernie Tomsa

Bernard Tomsa  
Attorney at Law  
Brooks Kushman P.C.  
1000 Town Center  
22nd Floor  
Southfield, MI 48075  
Phone: (248) 358-4400  
Fax: (248) 358-3351

CONFIDENTIALITY NOTICE: This message contains information that may be confidential and privileged. Unless you are the addressee or are authorized by the addressee, you may not use, copy or disclose the information contained in this message. If you have received this message in error, please delete it and advise the sender.

1. (Currently Amended) A computer readable storage medium storing one or more routines executable by a computer processor running on at least one collection computer, the routines comprising:

one or more load balancing routines that collect proportional server capability information for a plurality of servers, wherein the proportional server capability information is based at least in part on processing of sample requests transmitted to the plurality of servers during intervals;

wherein the one or more load balancing routines encode the collected proportional server capability information in a weighted distribution that represents the plurality of servers in a weighted manner based at least in part on the proportional server capability information, the weighted manner reflecting a percentage of total server capability realized by each server, wherein the encoded load balancing is stored in a persistent or non-persistent memory of the collection computer, wherein the encoding is one of either a linked list, a binary search tree, a hash table or an array, and wherein a proportion of entries, for each server, in the linked list, binary search tree, hash table or array correspond to the proportional server capability of each server; and

wherein the one or more routines further randomly select servers to process client requests from the weighted distribution.

2. (Cancelled).

3. (Previously Presented) The computer readable storage medium of claim 1 wherein the load balancing of client requests comprises selection of entries from the proportional server capability encoding, wherein each entry indicates at least one of the plurality of servers.

4-6. (Cancelled).

7. (Previously Presented) The computer readable storage medium of claim 1 wherein the collecting of server capability information comprises transmitting the sample requests to the plurality of servers during the intervals and recording information that corresponds to the servers servicing of the sample requests.

8. (Previously Presented) The computer readable storage medium of claim 1 wherein the sample requests include a mixture of configurable directory requests.

9. (Currently Amended) The computer readable storage medium of claim 1 wherein the server capability information includes one or more of ~~proportion~~ a percentage of serviced sample requests, time to serve each sample request, time to serve total sample requests, ~~proportion~~ a percentage of sample request types serviced, and time to serve sample request types.

10. (Previously Presented) The computer readable storage medium of claim 1 wherein the one or more load balancing routines update a proportional server capability based load balancing encoding in accordance with the collected server capability information.

11. (Previously Presented) The computer readable storage medium of claim 10 wherein the one or more load balancing routines update the proportional server capability based load balancing encoding in response to a change in network configuration.

12. (Previously Presented) The computer readable storage medium of claim 11 wherein the change of network configurations includes change of server availability.

13. (Previously Presented) The computer readable storage medium of claim 1 embodied in one or more of cache, registers, memory, and fast look-up tables.

14. (Currently Amended) A method comprising:  
during intervals, collecting data that reflects ~~proportional~~ capabilities of a plurality of backend servers at a collection computer, wherein the backend server capability data is based at least in part on servicing of sample requests by the plurality of backend servers;

encoding the collected backend server capability data to reflect ~~proportional~~ backend server capability of each of the plurality of backend servers, each server's percentage of the total processing capability being reflected, wherein the encoding is one of either a

linked list, a binary search tree, a hash table or an array, and wherein a proportion of entries, for each server, in the linked list, binary search tree, hash table or array correspond to the proportional server capability of each server; and

storing the encoded collected backend server capability data in at least one of a persistent or non-persistent memory included with the collection computer.

15. (Currently Amended) The method of claim 14 wherein the collected ~~proportional~~ backend server capability data is encoded to indicate each of the plurality of backend servers in accordance with their proportional capability based at least in part on the collected backend server capability data.

16. (Original) The method of claim 14 further comprising updating the encoding in accordance with the collected data.

17. (Original) The method of claim 14 further comprising load balancing client requests in accordance with the encoding.

18. (Original) The method of claim 17 wherein the load balancing comprises randomly selecting entries from the encoding, wherein the encoding includes entries that indicate the plurality of backend servers.

19-20. Cancelled.

21. (Currently Amended) The method of claim 14 wherein the collected backend server capability data includes one or more of: ~~proportion~~ a percentage of sample requests serviced by each of the backend servers, time for each of the backend servers to serve each sample request, time for each of the backend servers to serve total sample requests, proportion of sample request types serviced by each of the backend servers, and time for each of the backend servers to serve sample request types.

22. (Original) The method of claim 14 wherein collecting backend server capability data comprises:

transmitting the sample requests to the backend servers; and  
recording data that corresponds to servicing of the sample requests by the backend servers.

23. (Original) The method of claim 14 wherein the sample requests include sample directory requests.

24. (Original) The method of claim 23 wherein the sample directory requests are in accordance with one or more protocol including lightweight data access protocol and universal description discovery and integration.

25. (Previously Presented) The method of claim 14 executed by a computer processor reading a computer program product encoded on one or more machine-readable media.

26. (Currently Amended) A method comprising:  
maintaining a ~~proportional~~ server capability information encoding stored in at least one of a persistent or non-persistent memory in a computer, wherein the maintaining includes at least periodically updating the ~~proportional~~ server capability information stored in at least one of the persistent or non-persistent memory, and, wherein the encoding is one of either a linked list, a binary search tree, a hash table or an array, and wherein a proportion of entries, for each server, in the linked list, binary search tree, hash table or array correspond to the proportional server capability of each server;

load balancing client requests across a plurality of servers in accordance with the ~~proportional~~ server capability information encoding that reflects ~~proportional~~ capabilities of each the plurality of servers in relation to the capabilities of all the servers collectively, such that each server's percentage of the collective capability is reflected in the encoding, wherein the reflected ~~proportional~~ server capability information is based at least in part on servicing of sample requests by the plurality of servers.

27. (Original) The method of claim 26 wherein encoding reflects frequency of sample requests serviced by the servers.

28. (Original) The method of claim 27 wherein the frequency of sample requests serviced includes one or more of number of sample requests serviced during a time interval, number of sample requests serviced during a time interval based on type of sample requests, time to service a number of sample requests, and time to service a number of sample requests based on type of sample requests.

29. Cancelled.

30. (Currently Amended) The method of claim 26 further comprising collecting the ~~proportional~~ server capability information at intervals between servicing of client requests.

31. (Currently Amended) The method of claim 26 wherein updating the ~~proportional~~ server capability information encoding comprises:  
transmitting the sample requests to the plurality of servers at intervals; and  
recording the server capability information that indicates frequency of the sample requests serviced by the servers in the persistent or non-persistent memory.

32. (Original) The method of claim 31 wherein the sample requests include a mixture of configurable sample requests.

33. (Currently Amended) The method of claim 26 wherein the encoding includes a data structure that proportionally represents the plurality of servers in accordance with the ~~proportional~~ server capability information.

34. (Original) The method of claim 33 wherein the load balancing comprises selecting entries from the data structure at random.

35. (Original) The method of claim 34 wherein the load balancing comprises predetermined selection of entries from the data structure.

36. (Previously Presented) The method of claim 26 executed by a computer processor reading a computer program product encoded in one or more machine-readable medium.

37. (Currently Amended) A method comprising:

during a data collection interval,  
transmitting sample requests to servers,  
recording data that corresponds to servicing of the transmitted sample requests by each of the servers in at least one of a persistent or non-persistent memory in a recording computer; and encoding the recorded data, wherein the encoding of the data ~~proportionally~~ represents each of the servers in accordance with their ~~proportional~~ percentage of total server capability based at least in part on the recorded data, and  
wherein the encoding is one of either a linked list, a binary search tree, a hash table or an array, and wherein a proportion of entries, for each server, in the linked list, binary search tree, hash table or array correspond to the proportional server capability of each server.

38. (Currently Amended) The method of claim 37 wherein the encoding includes a load balancing table, and wherein a ~~proportion~~ number of entries for each server in the load balancing table exist in accordance with the ~~proportional~~ server capability of each server.

39. (Original) The method of claim 38 further comprising randomly selecting entries from the load balancing table to load balance client requests across the servers.

40. (Original) The method of claim 38 further comprising predetermined selection of entries from the load balancing table to load balance client requests across the servers.

41. (Original) The method of claim 37 further comprising load balancing client requests in accordance with the encoding.

42. (Original) The method of claim 37 further comprising randomly selecting entries from the load balancing structure to load balance client requests.

43. (Original) The method of claim 37 wherein the sample requests include search requests, compare requests, and update requests.

44. (Original) The method of claim 37 wherein the recorded data indicates one or more of number of sample requests serviced during the data collection interval by each of the directory servers, number of sample requests serviced during the data collection interval by each of the directory servers based on sample request type, time for each directory server to service a number of sample requests during the data collection interval, and time for each of the directory servers to service a number of sample requests based on type of sample requests during the data collection interval.

45. (Original) The method of claim 37 further comprising servicing client requests with a second plurality of servers during the data collection interval.

46. (Original) The method of claim 37 further comprising buffering client requests during the data collection interval.

47. (Previously Presented) The method of claim 37 executed by a computer processor reading a computer program product encoded in one or more machine-readable medium.

48-52. (Cancelled)

53. (Currently Amended) A machine-readable storage medium storing one or more instructions executable by a computer processor, the instructions comprising:

a first sequence of instructions to transmit sample requests to a plurality of servers at intervals and receive responses corresponding thereto;

a second sequence of instructions to determine ~~proportional~~ capability information for each of the plurality of servers that reflects ~~proportional~~ capabilities of each of the plurality of servers as a percentage of total server capability, the ~~proportional~~

capability being a ratio to the total capability of all the servers, based at least in part on the sample requests and corresponding responses; and

a third sequence of instructions to encode the determined ~~proportional~~ capability of each of the plurality of servers into a representative encoding, wherein the representative encoding represents each of the plurality of servers in accordance with each server's determined proportional capability, wherein the encoding is one of either a linked list, a binary search tree, a hash table or an array, and wherein a proportion of entries, for each server, in the linked list, binary search tree, hash table or array correspond to the proportional server capability of each server.

54. (Cancelled).

55. (Currently Amended) The machine readable storage medium of claim 53, wherein the third sequence of instructions further includes instructions to maintain the ~~proportional~~ server capability encoding for load balancing.

56. (Currently Amended) The machine readable storage medium of claim 53, further comprising a fourth sequence of instructions to load balance client requests in accordance with the ~~proportional~~ server capability encoding.

57. (Previously Presented) The machine readable storage medium of claim 56, further comprising a fifth sequence of instructions to buffer client requests during the intervals.

58. (Previously Presented) The machine readable storage medium of claim 56, further comprising the third sequence of instructions to forward client requests to standby servers during the intervals.

59. (Previously Presented) The machine readable storage medium of claim 53, wherein the second sequence of instructions measures one or more of time for each of a plurality of servers to respond to each request, time for each of a plurality of servers to

respond to each request based on request type, total number of responses provided by each of a plurality of servers during the periodic intervals, and number of responses provided by each of a plurality of servers based on request type.

60. (Currently Amended) A machine-readable storage medium, storing a computer program product comprising:

a first sequence of instructions to update a ~~proportional~~ server capability load balancing information encoding including indications for each server that reflect ~~proportional~~ measured sample request based capabilities of a plurality of servers, the reflection represented by relative occurrence of the indications for each server and indicative of a percentage of total server capability reflected by each server, wherein the encoding is one of either a linked list, a binary search tree, a hash table or an array, and wherein a proportion of entries, for each server, in the linked list, binary search tree, hash table or array correspond to the proportional server capability of each server,

wherein the proportional measured sample request based capabilities are measured during intervals; and

a second sequence of instructions to select server indications from the proportional server capability load balancing information encoding to load balance client requests.

61. (Currently Amended) The machine readable storage medium of claim 60, wherein the computer program product further comprises a third sequence of instructions to buffer client requests while the first sequence of instructions updates the ~~proportional~~ server capability load balancing information encoding.

62. (Currently Amended) The machine readable storage medium of claim 60, wherein the computer program product further comprises a third sequence of instructions to forward client requests to standby servers while the first sequence of instructions updates the ~~proportional~~ server capability load balancing information encoding.

63. (Previously Presented) The machine readable storage medium of claim 60, wherein the computer program product further comprises a third sequence of instructions to measure capabilities of the plurality of directory servers based at least in part on sample requests.

64. (Previously Presented) The machine readable storage medium of claim 63 wherein the third sequence of instructions to measure capabilities comprises the third sequence of instructions to transmit the sample requests to the plurality of directory servers and to receive responses corresponding thereto during the intervals.

65. (Currently Amended) The machine readable storage medium of claim 60 wherein the measured sample request based ~~proportional~~ capabilities include one or more of number of sample directory requests serviced during a time interval, number of sample directory requests serviced during a time interval based on type of sample requests, time to service a number of sample directory requests during a time interval, and time to service a number of sample directory requests based on type of sample requests during a time interval.

66. (Previously Presented) The machine readable storage medium of claim 60 wherein the requests include directory requests.

67. (Previously Presented) A network comprising:  
a plurality of servers processing requests; and  
a computer readable storage media containing one or more routines executable by a computer processor, the routines including one or more load balancing routines forwarding client requests in accordance with a ~~proportional~~ server capability information encoding that ~~proportionally~~ indicates each of the plurality of servers in accordance with their ~~proportional~~ processing capability as a percentage of total capability, wherein the ~~proportional~~ server capability information encoding is based at least in part on servicing of sample requests during intervals between forwarding of client requests, wherein the encoding is one of either a linked list, a binary search tree, a hash

table or an array, and wherein a proportion of entries, for each server, in the linked list, binary search tree, hash table or array correspond to the proportional server capability of each server.

68. (Previously Presented) The network of claim 67 wherein the one or more load balancing routines measuring the capabilities of the plurality of servers during the intervals.

69. (Previously Presented) The network of claim 68 wherein measuring includes transmitting the sample requests and receiving responses corresponding to the sample requests.

70. (Original) The network of claim 67 further comprising one or more standby servers to handle client requests while the load balancer measures capabilities of the plurality of servers.

71. (Currently Amended) A computer readable storage medium storing one or more routines executable by a computer processor, the execution of the one or more routines causing the computer processor to perform at least:

collecting ~~proportional~~ server capability information for a plurality of servers, wherein the server capability information is based at least in part on processing of sample requests transmitted to the plurality of servers;

encoding the ~~proportional~~ server capability information in a weighted distribution, such that portions of the weighted distribution corresponding to each server occupy a portion of the weighted distribution substantially equivalent to the ~~proportional~~ ~~server processing~~ capability of each of the servers as a percentage of the total capability, wherein the weighted distribution is further encoded as one of either a linked list, a binary search tree, a hash table or an array, and wherein a proportion of entries, for each server, in the linked list, binary search tree, hash table or array correspond to the ~~proportional~~ server capability of each server as a percentage of total server capability; and

selecting, from the weighted distribution, a server to process a client request.

72. (Currently Amended) The computer readable storage medium of claim 71, wherein the ~~proportional~~ server capability information is, representative of the server capability of each server in comparison to the server capability of all the servers.

73. (Previously Presented) The computer readable storage medium of claim 71, wherein the server capability information is based at least in part on processing of a single set of sample requests transmitted to the plurality of servers.

74. (Previously Presented) The computer readable storage medium of claim 71, wherein the sample requests are transmitted at regular intervals.

75. (Previously Presented) The computer readable storage medium of claim 71, wherein the selecting is done at random.

76. (Canceled).